

# Capítulo 6. Formatos e regras de EDIÇÃO

27 Mar 2025

## OBJETIVOS DO CAPÍTULO

- Conceitos e comandos de formatos de edição para variáveis do tipo caracter, inteiro e real
- Regras de edição de programas-fonte em FORTRAN
- Sequência obrigatória dos comandos dentro de programas-fonte em FORTRAN
- Novos comandos do FORTRAN a usar: WRITE(número,número), & e FORMAT

### 6.1 Formatos de EDIÇÃO para variáveis do tipo CARACTER: programa6a.f90

Para inicializar as atividades deste capítulo, deve-se:

- 1) Executar o programa Fortran
- 2) No Fortran, **criar um projeto** com o nome **programa06**
- 3) No Fortran, **criar e inserir** no projeto o programa-fonte **programa6a.f90**
- 4) Dentro do espaço de edição do Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 6.1**.

Tabela 6.1 Programa6a.f90.

```
USE PORTLIB

CHARACTER(10) A, B
CHARACTER(20) C
INTEGER VER

WRITE(*,*) "Variaveis do tipo caracter"
WRITE(*,*) "Entre com os conteudos de A e B"
READ(*,*) A, B

C = A // B

OPEN(9, file="saida6a.txt")

WRITE(9,*) A, "= A sem formato"

WRITE(9,21) A
21 FORMAT ( A2, "= A no formato A2" )

WRITE(9,22) A
22 FORMAT ( A5, "= A no formato A5" )
```

```

WRITE(9,23) A
23 FORMAT ( A10, "= A no formato A10" )

WRITE(9,33) A
33 FORMAT ( A, "= A no formato A-livre" )

WRITE(9,34) TRIM(ADJUSTL(A))
34 FORMAT ( A, "= A no formato A-livre mas com TRIM e ADJUSTL" )

WRITE(9,24) A
24 FORMAT("10X", 10X, 'A no formato A10 com "10X" e 10 espaços =(' , A10, ")" 1/)

WRITE(9,25) A
25 FORMAT("T10", T10, 'A no formato A10 com "T10" e tabulação 10=((' , A10, ")" , 1/)

WRITE(9,55) "Comentários no WRITE =(" , A, ")"
55 FORMAT(A, A, A1, /)

WRITE(9,10) A, B, C
10 FORMAT(5X, "A =", A10, 1/, 5X, "B =", A10, 1/, 5X, "C =", A20, 2/)

WRITE(9,11) A, B, C
11 FORMAT( 5X, "A =", A10, 1/,      &
           5X, "B =", A10, 1/,      &
           5X, "C =", A20, //      )

WRITE(9,12) A, B
12 FORMAT ( 2(A10), "= variáveis A e B no formato 2(A10)" )

CLOSE(9)

VER = SYSTEM("Notepad saida6a.txt")

END

```

5) Comentários sobre o programa:

- O objetivo do programa6a.f90 é explicar alguns formatos de edição utilizados com variáveis do tipo caracter.
- No capítulo 5, além de um asterisco, o comando WRITE foi utilizado com um número para identificar o “local” (janela ou arquivo) onde seria escrito algo, isto é, ele foi usado na forma WRITE(número1,\*). Um exemplo é a linha WRITE(9,\*) A, "= A sem formato" deste programa. Agora, no programa6a.f90, o comando WRITE é utilizado com dois números, ou seja, na forma WRITE(número1,número2), por

exemplo, a linha **WRITE(9,21) A** do programa. O número2 está associado a um novo comando do FORTRAN chamado de FORMAT.

- c) O comando FORMAT é usado para especificar o formato com que comentários e conteúdos (strings ou números) de variáveis são escritos nos dispositivos de saída, seja uma janela ou um arquivo. Também é usado na leitura de dados.
- d) A sintaxe, isto é, a forma com que se deve usar o comando FORMAT é: número2 FORMAT (formatos). O número2 deve ser o mesmo utilizado no respectivo comando WRITE. Dentro de parênteses, deve-se especificar os formatos de edição que serão usados, separados por vírgula. Um exemplo é a linha **21 FORMAT ( A2, "= A no formato A2" )** do programa, que especifica o formato A2 para escrever a variável A, do tipo caracter, chamada na linha **WRITE(9,21) A** do programa.
- e) Os comandos FORMAT podem ser incluídos em qualquer lugar dentro do programa-fonte, desde que seja abaixo de comandos USE e acima do comando END. Eles podem ser aglomerados em um único lugar ou cada um pode estar abaixo ou acima do seu respectivo comando WRITE. Não é necessário seguir qualquer ordenamento para os números usados nos comandos FORMAT, nem para suas posições relativas dentro do programa-fonte. E não existe qualquer relação entre os números usados em comandos OPEN e os números usados nos comandos FORMAT. Eles até podem ser iguais. Porém, dois comandos FORMAT não podem ter o mesmo número.
- f) Cada tipo de variável (caracter, inteiro ou real) tem identificadores especiais para escrever seu conteúdo na forma desejada. Além disso, existem identificadores que facilitam a escrita de comentários e da forma ou disposição estética com que se escreve algo.
- g) O identificador de variáveis do tipo caracter é a letra A seguida de um número que especifica a quantidade máxima de caracteres para escrever cada variável. Dentro deste espaço, o conteúdo de cada variável do tipo caracter é escrito da esquerda para a direita. Se o conteúdo da variável tem mais caracteres do que o reservado para escrevê-la, a parte excedente não é escrita, é desconsiderada. Um exemplo de uso deste identificador é A2, citado no item d, acima. Neste caso, reservou-se duas colunas ou caracteres para escrever uma variável do tipo caracter.
- h) Dentro de um comando FORMAT, isto é, entre parênteses, pode-se escrever um ou mais comentários, cada um devendo ficar entre aspas ou apóstrofos, e separados por vírgula, como na linha  
**22 FORMAT ( A5, "= A no formato A5" )** do programa.
- i) O identificador X é usado para deixar espaços em branco entre dois formatos de edição. Ele é usado na forma NX, onde N é um número que representa a quantidade de espaços em branco. Um exemplo é a linha **24 FORMAT("10X", 10X, 'A no formato A10 com "10X" e 10 espaços =(' , A10, ")" 1/)** do programa, onde foi usado o formato 10X. Deve-se perceber que "10X" é apenas um comentário.
- j) Nesta mesma linha do programa há o formato de edição N/, onde N é um número que representa a quantidade de linhas que o cursor descerá quando encontrar este formato. Neste caso, uma linha, devido ao formato 1/. Também pode-se usar várias / em vez de N/, como mostrado no 11 FORMAT.

- k) O identificador T, de tabulação, é usado para escrever algo a partir de uma coluna específica, onde a coluna 1 fica na margem esquerda. Ele é usado na forma TN, onde N é o número da colunas a partir da qual se escreverá algo que vem a seguir dentro de um comando FORMAT. Um exemplo é a linha `25 FORMAT("T10", T10, 'A no formato A10 com "T10" e tabulação 10=(', A10, ")", 1/)` do programa, onde foi usado o formato T10. Deve-se perceber que "T10" é apenas um comentário.
- l) Pode-se usar um único comando WRITE, com seu respectivo comando FORMAT, para escrever diversas variáveis. Neste caso, cada variável será escrita com o formato que estiver na mesma sequência dos comandos WRITE e FORMAT. Por exemplo, na linha `WRITE(9,10) A, B, C` do programa, são escritas as variáveis A, B e C com o formato número 10, que corresponde à linha `10 FORMAT(5X, "A =", A10, 1/, 5X, "B =", A10, 1/, 5X, "C =", A20, 2/)` do programa. Portanto, a primeira variável (A) será escrita com o formato A10, que é o primeiro formato de edição de variáveis que aparecem no comando FORMAT. E as variáveis B e C serão escritas com os formatos A10 e A20, respectivamente, que são o segundo e o terceiro formatos de variáveis no comando FORMAT.
- m) O comando FORMAT identificado pelo número 10 no programa tem exatamente a mesma função que o comando FORMAT identificado pelo número 11. A única diferença é na edição do programa-fonte. No 10 FORMAT, todos os formatos de edição usam apenas uma linha de edição do programa-fonte, ou seja, a instrução ocupa apenas uma linha. Já no 11 FORMAT, são usadas três linhas de edição para uma única instrução. Neste caso, quando um mesmo comando precisa ou se quer empregar mais de uma linha de edição por motivos estéticos, usa-se o símbolo "&", ao final de cada linha do programa-fonte, para indicar que a instrução continua na linha seguinte. Isso vale para qualquer comando do FORTRAN.
- n) O formato M(AN) é equivalente a M vezes o formato AN, onde N é o número de caracteres reservados para se escrever uma variável do tipo caracter. Por exemplo, na linha `12 FORMAT ( 2(A10), "= variáveis A e B no formato 2(A10) " )` do programa usa-se duas vezes o formato A10 para que duas variáveis do tipo caracter sejam escritas.
- o) Deve-se notar que em todos os exemplos deste programa, exceto em um, os comentários ficam dentro dos comandos FORMAT. Mas isso não é obrigatório, como mostrado na linha `WRITE(9,55) "Comentários no WRITE =(", A, ")".` Mas, neste caso, deve-se perceber que também é necessário usar o identificador A para os comentários, como em `55 FORMAT(A, A, A1, /).`
- p) Os comandos vistos no Capítulo 4 para variáveis do tipo caracter também podem ser usados nos comandos WRITE associados a comandos FORMAT, como em `WRITE(9,34) TRIM(ADJUSTL(A)).`
- q) Cada novo comando WRITE faz o cursor iniciar em uma nova linha no dispositivo de saída.
- r) É possível usar o identificador A sem especificar o número de caracteres, como mostrado na linha `33 FORMAT ( A, "= A no formato A-livre" ).`
- 6) Executar **Build, Compile** para compilar o programa.
- 7) Gerar o programa-executável fazendo **Build, Build**.
- 8) Ao se executar o programa, através de **Build, Execute**, surge uma janela do DOS, mostrada na Figura 6.1, dentro da qual tem-se:

a) Na primeira linha, o comentário “Variaveis do tipo caracter”, resultado do comando

```
WRITE(*,*) "Variaveis do tipo caracter"
```

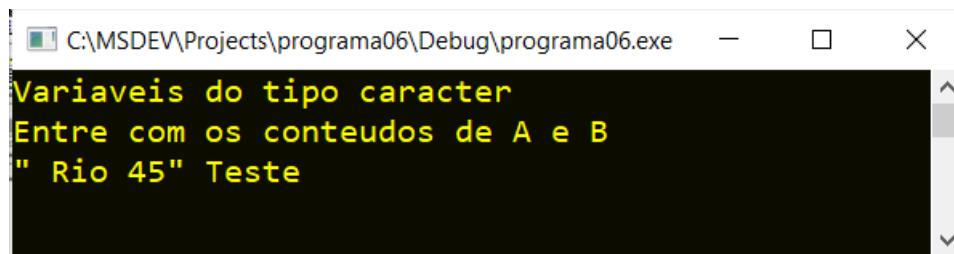


Figura 6.1 Janela DOS do programa6a.f90.

b) Na segunda linha, o comentário “Entre com os conteudos de A e B”, resultado do comando

```
WRITE(*,*) "Entre com os conteudos de A e B"
```

c) Na terceira linha, o programa pára e fica aguardando até que sejam fornecidos os conteúdos das variáveis A e B, resultado do comando `READ(*,*) A, B` do programa. Para que o programa continue a sua execução é necessário **digitar “ Rio 45” Teste**, por exemplo, e, em seguida, **clicar na tecla enter**.

```
Rio 45 = A sem formato
R= A no formato A2
Rio = A no formato A5
Rio 45 = A no formato A10
Rio 45 = A no formato A-livre
Rio 45= A no formato A-livre mas com TRIM e ADJUSTL
10X A no formato A10 com "10X" e 10 espaços =( Rio 45 )

T10 A no formato A10 com "T10" e tabulação 10=( Rio 45 )

Comentários no WRITE =( Rio 45 )
A = Rio 45
B =Teste
C = Rio 45 Teste

A = Rio 45
B =Teste
C = Rio 45 Teste

Rio 45 Teste = variáveis A e B no formato 2(A10)
```

Figura 6.2 Arquivo saida6a.txt do programa6a.f90.

- d) Em seguida ocorrerá a criação do arquivo saída6a.txt, dentro da pasta do projeto, como resultado do comando `OPEN(9, file="saída6a.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa6a.f90 for executado pela primeira vez. Este comando também declara que será usado o número 9 no programa como referência ao arquivo saída6a.txt.
- e) Como resultado da linha `VER = SYSTEM("Notepad saída6a.txt")` do programa, o aplicativo Notepad abre automaticamente o arquivo saída6a.txt cujo conteúdo é mostrado na Figura 6.2. Na segunda e terceira linhas do arquivo saída6a.txt, o conteúdo da variável A foi cortado parcialmente devido aos poucos caracteres usados para escrevê-la, no caso 2 e 5, definidos por A2 e A5. Para evitar este tipo de problema, sempre se deve superdimensionar o espaço para escrever o conteúdo de cada variável, não usar formato ou usar o formato-livre.
- 9) Até entender, **comparar** o resultado de cada comando WRITE, e seu respectivo comando FORMAT no programa6a.f90, mostrado em cada linha da Figura 6.2 ou no arquivo saída6a.txt.
- 10) **Executar** novamente o programa com outros conteúdos para as variáveis A e B. **Analizar** os novos resultados.

## 6.2 Formatos de EDIÇÃO para variáveis do tipo INTEIRO: programa6b.f90

Nesta seção será usado o mesmo projeto da seção anterior. Portanto, deve-se executar o seguinte no Fortran:

- 1) **Clicar** sobre o nome do programa-fonte
- 2) **Edit, Cut** para retirar o programa-fonte do projeto.
- 3) **Criar e inserir** no projeto o programa-fonte **programa6b.f90**
- 4) Dentro do espaço de edição do Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 6.2**.
- 5) Comentários sobre o programa:
  - a) O objetivo do programa6b.f90 é explicar alguns formatos de edição utilizados com variáveis do tipo inteiro.
  - b) No caso do programa6b.f90, os comandos FORMAT foram todos aglomerados no final do programa-fonte, na mesma ordem de seus respectivos comandos WRITE, mas isto não é necessário.
  - c) O identificador de variáveis do tipo inteiro é a letra I seguida de um número que especifica a quantidade máxima de algarismos ou colunas para escrever o valor de cada variável. Dentro deste espaço, o conteúdo de cada variável do tipo inteiro é escrito da direita para a esquerda. Uma coluna é usada para o sinal do número inteiro. Um exemplo de uso deste identificador é I5 na linha  
`22 FORMAT ( I5, "= A no formato I5" )` do programa.
  - d) Se o valor inteiro da variável tiver mais algarismos do que o reservado para escrevê-la, no lugar do número serão apresentados apenas asteriscos. Isso significa que o formato utilizado é incompatível com o valor da variável. Um exemplo disso é mostrado na segunda linha da Figura 6.4 que corresponde à seguinte instrução do programa: `21 FORMAT ( I2, "= A no formato I2" )`.

- e) Os identificadores X, para espaços em branco, T, para tabulação, e N/, para o cursor descer N linhas, explicados na seção 6.1 para variáveis do tipo caracter, também se aplicam a variáveis do tipo inteiro, conforme pode-se ver no programa6b.f90.

Tabela 6.2 Programa6b.f90.

```

USE PORTLIB

INTEGER A, B, C
INTEGER VER

WRITE(*,*) "Variaveis do tipo inteiro"
WRITE(*,*) "Entre com os valores de A e B"
READ(*,*) A, B

C = A + B

OPEN(9, file="saida6b.txt")

WRITE(9,*) A, "= A sem formato"
WRITE(9,21) A
WRITE(9,22) A
WRITE(9,23) A
WRITE(9,24) A
WRITE(9,25) A
WRITE(9,10) A, B, C
WRITE(9,11) A, B, C
WRITE(9,9) A, B, C

CLOSE(9)

VER = SYSTEM("Notepad saida6b.txt")

21 FORMAT ( I2, "= A no formato I2" )
22 FORMAT ( I5, "= A no formato I5" )
23 FORMAT ( I10, "= A no formato I10" )
24 FORMAT("10X", 10X, "Valor de A no formato I10 =", I10, 1/)
25 FORMAT("T10", T10, "Valor de A no formato I10 =", I10, 1/)
10 FORMAT(5X, "A =", I10, 1/, 5X, "B =", I10, 1/, 5X, "C =", I10, 2/)
11 FORMAT( 5X, "A =", I10, 1/,      &
           5X, "B =", I10, 1/,      &
           5X, "C =", I10, 2/      )
9 FORMAT ( 3(I10), "= variáveis A, B e C no formato 3(I10)" )

END

```

- f) O formato M(IN) é equivalente a M vezes o formato IN, onde N é o número de colunas ou algarismos reservados para escrever uma variável do tipo inteiro. Por exemplo, na linha `9 FORMAT ( 3(I10) , "= variáveis A, B e C no formato 3(I10)" )` do programa, o formato I10 é usado três vezes para escrever três variáveis do tipo inteiro.
- 6) Executar **Build, Compile** para compilar o programa.
- 7) Gerar o programa-executável fazendo **Build, Build**.
- 8) Ao se executar o programa, através de **Build, Execute**, surge uma janela do DOS, mostrada na Figura 6.3, dentro da qual tem-se:

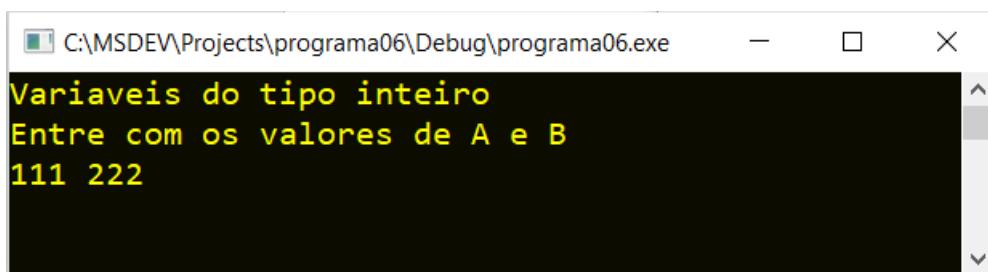
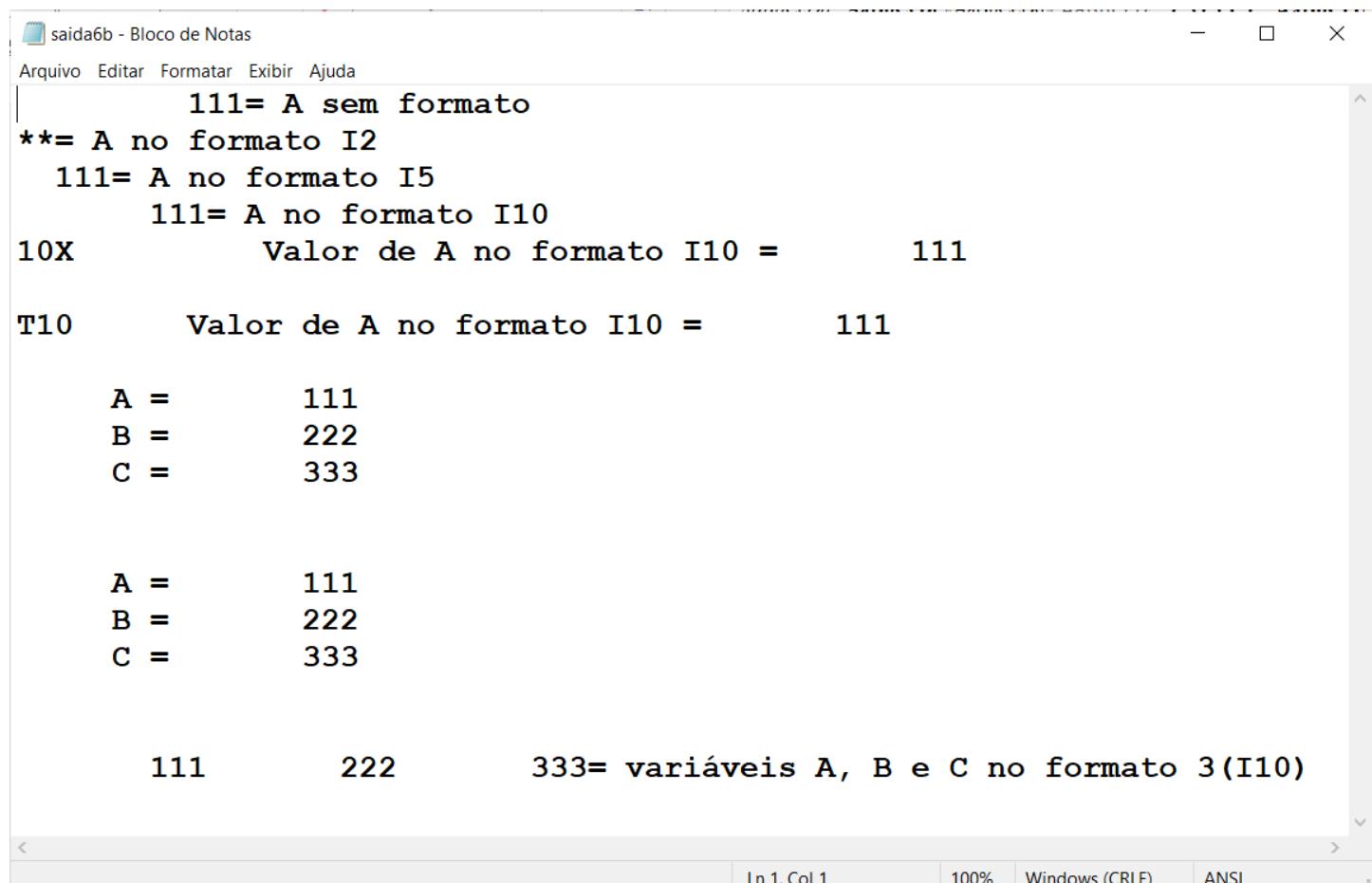


Figura 6.3 Janela DOS do programa6b.f90.

- a) Na primeira linha, o comentário “Variaveis do tipo inteiro”, resultado do comando `WRITE(*,*) "Variaveis do tipo inteiro"` do programa.
- b) Na segunda linha, o comentário “Entre com os valores de A e B”, resultado do comando `WRITE(*,*) "Entre com os valores de A e B"` do programa.
- c) Na terceira linha, o programa pára e fica aguardando até que sejam fornecidos os valores das variáveis A e B, resultado do comando `READ(*,*) A, B` do programa. Para que o programa continue a sua execução é necessário **digitar 111 222**, por exemplo, e, em seguida, **clicar na tecla enter**.
- d) Em seguida ocorrerá a criação do arquivo saída6b.txt, dentro da pasta do projeto, como resultado do comando `OPEN(9, file="saída6b.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa6b.f90 for executado pela primeira vez. Este comando também declara que será usado o número 9 no programa como referência ao arquivo saída6b.txt.
- e) Como resultado da linha `VER = SYSTEM("Notepad saída6b.txt")` do programa, o aplicativo Notepad abre automaticamente o arquivo saída6b.txt cujo conteúdo é mostrado na Figura 6.4. Na segunda linha do arquivo saída6b.txt o valor da variável A (111) foi trocado por dois asteriscos. Isso ocorre porque o formato usado (I2) é incompatível para se escrever o valor da variável A, que neste caso precisa de pelo menos três colunas, ou seja, o formato mínimo seria I3. Para evitar este tipo de problema, sempre se deve superdimensionar o espaço para escrever o valor de cada variável.
- 9) Até entender, **comparar** o resultado de cada comando WRITE, e seu respectivo comando FORMAT no programa6b.f90, mostrado em cada linha da Figura 6.4 ou no arquivo saída6b.txt.

10) **Executar** novamente o programa com outros valores para as variáveis A e B e **analisar** os novos resultados; usar por exemplo:

- a) **Primeiro caso: 11 e 222**
- b) **Segundo caso: -11 e 222**



```
saída6b - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
111= A sem formato
**= A no formato I2
111= A no formato I5
111= A no formato I10
10X      Valor de A no formato I10 =      111
T10      Valor de A no formato I10 =      111
A =      111
B =      222
C =      333
A =      111
B =      222
C =      333
111      222      333= variáveis A, B e C no formato 3(I10)
```

Figura 6.4 Arquivo saída6b.txt do programa6b.f90.

### 6.3 Formatos de EDIÇÃO para variáveis do tipo REAL: programa6c.f90

Nesta seção será usado o mesmo projeto da seção anterior. Portanto, deve-se executar o seguinte no Fortran:

- 1) **Clicar** sobre o nome do programa-fonte
- 2) **Edit, Cut** para retirar o programa-fonte do projeto.
- 3) **Criar e inserir** no projeto o programa-fonte **programa6c.f90**
- 4) Dentro do espaço de edição do Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 6.3**.
- 5) Comentários sobre o programa:
  - a) O objetivo do programa6c.f90 é explicar alguns formatos de edição utilizados com variáveis do tipo real.
  - b) No caso do programa6c.f90, os comandos FORMAT foram todos aglomerados no início do programa-fonte, antes mesmo de seus respectivos comandos WRITE.
  - c) Podem ser usados dois identificadores com variáveis do tipo real. Eles são representados pelas letras F e E.

d) Na notação decimal, usa-se o formato de ponto flutuante (F) através de FX.Y, onde Y é um número que equivale à quantidade de algarismos usados para representar a parte decimal do valor da variável, e X é um número que equivale à quantidade total de colunas usadas para representar o valor da variável, incluindo seu sinal, o ponto decimal e as partes inteira e decimal; portanto, o número X sempre tem que ser maior do que Y. No programa, por exemplo, são usados os formatos F5.2 e F5.0.

Tabela 6.3 Programa6c.f90.

```

USE PORTLIB

REAL A, B, C
INTEGER VER

11 FORMAT( 5X, "A =", 1PE15.3, 1/,      &
           T5, "B =", 1PE15.3, 1/,      &
           5X, "C =", 1PE15.3, 1/      )

12 FORMAT ( 3(1PE15.3), "= variáveis A, B e C no formato 3(1PE15.3) " )

21 FORMAT ("A no formato F5.2      =", F5.2,      1/, &
           "A no formato F5.0      =", F5.0,      1/, &
           "A no formato E15.3     =", E15.3,     1/, &
           "A no formato 1PE15.3   =", 1PE15.3,   1/  )

WRITE(*,*) "Variaveis do tipo real"
WRITE(*,*) "Entre com os valores de A e B"
READ(*,*) A, B

C = A + B

OPEN(9, file="saida6c.txt")

WRITE(9,*) A, "= A sem formato"
WRITE(9,21) A, A, A
WRITE(9,11) A, B, C
WRITE(9,12) A, B, C

CLOSE(9)

VER = SYSTEM("Notepad saida6c.txt")

END

```

e) Na notação científica, usa-se o formato exponencial (E) através de EX.Y, onde Y é um número que equivale à quantidade de algarismos usados para representar a parte decimal do valor da variável, X é um número que equivale à quantidade total de colunas usadas para representar o valor da variável, incluindo seu sinal,

o ponto decimal, a parte decimal, a letra E, o sinal do expoente e dois algarismos para o valor do expoente; portanto, o número X sempre tem que ser maior do que Y. No programa, por exemplo, é usado o formato E15.3.

- f) No formato EX.Y, nenhum algarismo é usado na parte inteira. Para que seja apresentado um algarismo na parte inteira do número real, pode-se utilizar o formato 1PEX.Y. No programa, por exemplo, é usado o formato 1PE15.3.
- g) Dentro das X colunas dos formatos FX.Y, EX.Y e 1PEX.Y, o conteúdo de cada variável do tipo real é escrito da direita para a esquerda. Os valores de X e Y são definidos pelo programador.
- h) Os identificadores X, para espaços em branco, T, para tabulação, e N/, para o cursor descer N linhas, explicados na seção 6.1 para variáveis do tipo caracter, também se aplicam a variáveis do tipo real, conforme pode-se ver no programa6c.f90.
- i) O formato M(1PEX.Y) é equivalente a M vezes o formato 1PEX.Y. Por exemplo, na linha
- ```
12 FORMAT ( 3(1PE15.3), "= variáveis A, B e C no formato 3(1PE15.3) " )
```
- do programa, usa-se três vezes o formato 1PE15.3 para escrever três variáveis do tipo real.

- 6) Executar **Build, Compile** para compilar o programa.
- 7) Gerar o programa-executável fazendo **Build, Build**.
- 8) Ao se executar o programa, através de **Build, Execute**, surge uma janela do DOS, mostrada na Figura 6.5, dentro da qual tem-se:

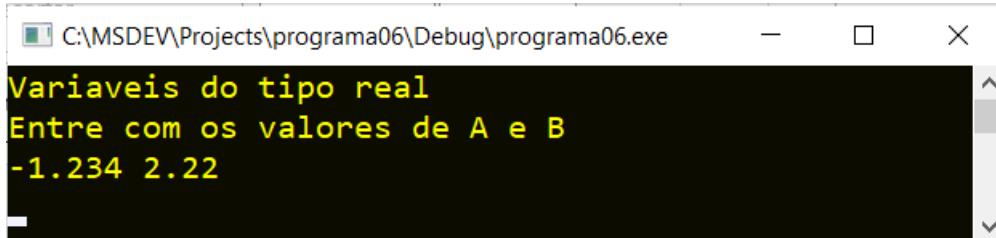


Figura 6.5 Janela DOS do programa6c.f90.

- a) Na primeira linha, o comentário “Variaveis do tipo real”, resultado do comando

```
WRITE(*,*) "Variaveis do tipo real"
```

do programa.- b) Na segunda linha, o comentário “Entre com os valores de A e B”, resultado do comando

```
WRITE(*,*) "Entre com os valores de A e B"
```

do programa.- c) Na terceira linha, o programa pára e fica aguardando até que sejam fornecidos os valores das variáveis A e B, resultado do comando **READ(\*,\*) A, B** do programa. Para que o programa continue a sua execução é necessário **digitar -1.234 2.22**, por exemplo, e, em seguida, **clicar na tecla enter**.
- d) Em seguida ocorrerá a criação do arquivo saída6c.txt, dentro da pasta do projeto, como resultado do comando **OPEN(9, file="saída6c.txt")** do programa, uma vez que este arquivo ainda não existe quando o programa6c.f90 for executado pela primeira vez. Este comando também declara que será usado o número 9 no programa como referência ao arquivo saída6c.txt.

- e) Como resultado da linha `VER = SYSTEM("Notepad saida6c.txt")` do programa, o aplicativo Notepad abre automaticamente o arquivo saida6c.txt cujo conteúdo é mostrado na Figura 6.6.
- f) Dependendo do formato empregado, ocorre um arredondamento do número, isto é, há perda de um ou mais algarismos do número real original (dado), como se pode ver da segunda à quarta linhas do arquivo saida6c.txt e com o próximo exemplo.

```

saida6c - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
-1.234000= A sem formato
A no formato F5.2 ==-1.23
A no formato F5.0 = -1.
A no formato E15.3 = -.123E+01
A no formato 1PE15.3 = -1.234E+00

A = -1.234E+00
B = 2.220E+00
C = 9.860E-01

-1.234E+00 2.220E+00 9.860E-01= variáveis A, B e C no formato 3(1PE15.3)

```

Figura 6.6 Arquivo saida6c.txt do programa6c.f90.

- 9) Até entender, **comparar** o resultado de cada comando WRITE, e seu respectivo comando FORMAT no programa6c.f90, mostrado em cada linha da Figura 6.6 ou no arquivo saida6c.txt.
- 10) **Executar** novamente o programa com outros valores para as variáveis A e B, e **analisar** os novos resultados. **Usar**, por exemplo, **-1.987 e 0.51**.
- 11) Um mesmo comando FORMAT pode ter identificadores de variáveis do tipo caracter, inteiro e real, além de comentários, e dos formatos NX, N/ e TN. Isto será visto em vários exemplos em capítulos futuros.

#### **6.4 Resumo de REGRAS GERAIS DE EDIÇÃO de programas em FORTRAN**

- 1) Em cada linha do programa, tudo que estiver após o símbolo de exclamação (!) não é executado pelo programa. Serve apenas como um comentário, por exemplo, para esclarecer o que faz uma determinada parte do programa. Dentro do Fortran PowerStation 4.0, todos os comentários ficam na cor verde. Linhas em branco são equivalentes a ter o símbolo (!) de comentário na primeira coluna de cada linha em branco.
- 2) Tudo que é criado ou definido pelo programador fica na cor preta dentro do Fortran PowerStation 4.0.
- 3) Todos os comandos intrínsecos da linguagem FORTRAN ficam na cor azul dentro do Fortran PowerStation 4.0.
- 4) O símbolo & é usado ao final de uma linha de programa para indicar que, a instrução iniciada nesta linha de edição, continua na linha seguinte.
- 5) Cada linha do programa pode usar até 132 colunas, incluindo os espaços em branco.

- 6) Em geral, espaços em branco têm apenas função estética no programa, não afetando os comandos ou as instruções do programador. Mas, os espaços em branco são considerados quando usados dentro de comentários nos comandos WRITE e FORMAT.
- 7) Nos comandos da linguagem FORTRAN e em nomes de variáveis, não existe distinção entre letras maiúsculas e minúsculas. Por exemplo, WRITE, write, Write e wRiTé são equivalentes. A distinção entre letras minúsculas e maiúsculas existe dentro de comentários nos comandos WRITE e FORMAT.
- 8) Na linguagem FORTRAN, os nomes de variáveis podem ter até 31 caracteres compostos pela combinação das 26 letras do alfabeto, dos algarismos 0, 1, 2, 3 a 9, e o traço de sublinhado. Não é permitido acentuação e todas as variáveis devem começar por uma letra. Exemplos de nomes válidos para variáveis (separadas por vírgula): v, va, variavel, variavel\_2, variavel\_31, nome03, nome\_03, nome\_da\_rua.
- 9) Todas as variáveis que iniciam pelas letras I, J, K, L, M e N são consideradas do tipo inteiro se não forem declaradas como reais ou caracteres. E, todas as variáveis que iniciam pelas demais letras do alfabeto são consideradas do tipo real se não forem declaradas como inteiras ou caracteres.

## **6.5 SEQUÊNCIA obrigatória de comandos em FORTRAN**

Até este capítulo foram vistos comandos da linguagem FORTRAN que podem ser divididos em cinco tipos:

- 1) Comando para inclusão de biblioteca no programa-fonte: USE
- 2) Comandos para declarar tipos de variáveis: INTEGER, REAL e CHARACTER
- 3) Outros comandos intrínsecos (WRITE, READ, OPEN, CLOSE) ou não do FORTRAN (SYSTEM), funções intrínsecas (LOG, TRIM etc) e qualquer outro comando criado pelo programador, por exemplo, cálculos e operações com strings
- 4) Comando de edição: FORMAT
- 5) Comando de encerramento do programa-fonte: END

Se um programa-fonte tiver todos os cinco tipos de comandos da lista acima, a sequência obrigatória entre eles é exatamente a apresentada. Porém, o comando FORMAT pode ficar em qualquer lugar do programa-fonte entre os comandos USE e END.

Para os comandos de declaração de variáveis (INTEGER, REAL e CHARACTER), a ordem entre eles pode ser qualquer. O mesmo se aplica para os comandos intrínsecos, exceto com arquivos pois, para um mesmo arquivo, a sequência obrigatória é: OPEN, WRITE e outros comandos, CLOSE.

## 6.6 EXERCÍCIOS

### Exercício 6.1

Editar um programa-fonte em FORTRAN para realizar um cálculo de interpolação linear executando o seguinte algoritmo (passos):

- 1) Ler os valores de duas variáveis do tipo real, chamadas X1 e X2, que correspondem a dois dados tabelados de uma variável independente
- 2) Ler os valores de duas variáveis do tipo real, chamadas Y1 e Y2, que correspondem a dois dados tabelados de uma variável dependente de X
- 3) Ler o valor de uma variável do tipo real, chamada X, para o qual se deseja realizar a interpolação
- 4) Calcular o valor da variável Y, em função do valor de X, usando interpolação linear
- 5) Utilizando comandos FORMAT, escrever no arquivo chamado saída\_6p1.txt os cinco valores lidos e o resultado de Y, juntamente com comentários para identificá-los
- 6) Executar o programa para X1 = 10, X2 = 20, Y1 = 100, Y2 = -20 e X = 12. O resultado deve ser Y = 76.

### Exercício 6.2

Editar um programa-fonte em FORTRAN para realizar cálculos com uma progressão geométrica (P.G.) executando o seguinte algoritmo (passos):

- 1) Ler o primeiro valor (real) da P.G., denotado por a1
- 2) Ler a razão (número real) entre dois termos subseqüentes da P.G., denotada por q
- 3) Ler o número (inteiro) de termos da P.G., denotado por N
- 4) Calcular o valor (real) do último termo da P.G., denotado por an, com  $an = a1 * q^{**}(N-1)$
- 5) Calcular a soma de todos os termos da P.G., denotada por Sn, com  $Sn = a1 * (q^{**}N - 1)/(q - 1)$
- 6) Utilizando comandos FORMAT, escrever no arquivo chamado saída\_6p2.txt os três valores lidos e os dois calculados, juntamente com comentários para identificá-los
- 7) Executar o programa para a1 = 1.5, q = 1.1 e N = 20. Os resultados devem ser aproximadamente  $an \approx 9.174$  e  $Sn \approx 85.91$ .

### **Exercício 6.3**

- 1) Ler a idade de uma pessoa em anos (variável do tipo inteiro) e seu nome (caracter)
- 2) Calcular (variáveis reais) o tempo total de vida da pessoa, expressando o mesmo resultado em 4 unidades diferentes:
  - a) dias
  - b) horas
  - c) minutos
  - d) segundos
- 3) Usando formatos de edição adequados, escrever em arquivo os dados e resultados, junto com comentários para identificá-los
- 4) Abrir automaticamente o arquivo com o NOTEPAD

### **Exercício 6.4**

- 1) Criar um projeto com o nome **Cap6p4** e inserir nele o programa-fonte **Cap6p4.f90**
- 2) Ler o nome do aluno e o ano em que estamos, e atribuí-los às variáveis Nome (caracter) e Ano (inteiro)
- 3) Calcular o valor de  $\pi$  com  $\text{acos}(-1.0)$  e atribuí-lo à variável real Pi
- 4) Escrever as variáveis Nome e Ano no arquivo de saída **Cap6p4.TXT** usando o comando format e o nome das variáveis
- 5) Escrever a variável Pi no arquivo de saída usando o comando format de ponto flutuante com 2 algarismos decimais, e comentários com o nome da variável e o formato usado
- 6) Escrever a variável Pi no arquivo de saída usando o comando format de notação científica com 1 algarismo na parte inteira e 2 algarismos decimais, e comentários com o nome da variável e o formato usado
- 7) Abrir o arquivo de saída com o aplicativo Bloco de Notas (Notepad).